

Editor de Texto Acessível para Web

Daniela Bagatini
UNISC/FADERGS/UFRGS
bagatini@unisc.br

Thiago Eduardo Viganico
UNISC/PROCERGS
thiagovi@unisc.br

Márcia E.J. Kniphoff Da Cruz
UNISC
mcruz@unisc.br

Rolf F. Molz
UNISC
rolf@unisc.br

RESUMO

A partir do desenvolvimento de páginas e sistemas Web, a Internet passou a caracterizar-se como um instrumento indispensável na busca de novos conhecimentos. Entretanto, em algumas situações os recursos oferecidos na Internet não contemplam a acessibilidade para todos. Há pessoas que têm necessidades especiais e que, seguidamente, apresentam dificuldades de aproveitamento desses recursos. Com o objetivo de reduzir as barreiras são apresentadas novas funcionalidades para um editor de textos acessível com base nas Recomendações de Acessibilidade para Conteúdo Web (WCAG 2.0). Para identificar e validar recursos úteis para o editor foram realizados testes de usabilidade com usuários cegos. O produto deste trabalho é disponibilizado como software livre e pode ser utilizado por grupos de pesquisas na área de acessibilidade.

Categories and Subject Descriptors

H.5.2 [User Interfaces]: Evaluation/methodology.

General Terms

Measurement, Verification, Experimentation.

ABSTRACT

Since of the development of many pages and web systems, the Internet has been featured as a necessary instrument to new knowledge search. However, at some situations the Internet has offered resources that do not cover the accessibility to all. There are people who have special needs and they find it difficult to use these resources. With the goal of reducing these limits, it is presented the new features to an Accessible Text Editor, based on the Web Content Accessibility Guidelines (WCAG 2.0). To identify and validate useful features to the editor were conducted usability test with blind users. The result of this job is provided as free software and it can be used by research groups about the accessibility.

Keywords

Accessibility, Accessible Text Editor, Usability Test.

1. 1. INTRODUÇÃO

O conceito de inclusão social promoveu a inserção das pessoas com deficiência (PcD) no mercado de trabalho, na educação básica e no ensino superior. Algumas iniciativas do Governo Federal Brasileiro como Lei de Cotas (Lei 8.213, de 24 de julho de 1991)- que obriga as empresas públicas e privadas a destinar um percentual das vagas de trabalho para estas pessoas, a gradual extinção das classes especiais nas escolas públicas que eram destinadas às crianças com necessidades educacionais especiais (NEE's)-, e a inclusão das crianças com NEE's nas salas de aula

regulares (Lei de Diretrizes e Bases da Educação Nacional 9.394, 1996, art. 58); estão contribuindo para a inclusão das pessoas com necessidades especiais na sociedade [1].

Com o mesmo propósito, o Consórcio World Wide Web (W3C - World Wide Web Consortium) criou a Iniciativa de Acessibilidade na Web (WAI - Web Accessibility Initiative) para discutir e arquitetar ações para o favorecimento da acessibilidade na Web. O consórcio publicou as Recomendações para Conteúdo Acessível na Web (WCAG - Web Content Accessibility Guidelines), que reúne diversas considerações que visam tornar o conteúdo Web acessível. Segundo a pesquisa realizada pelo W3C.BR/NIC.br intitulada "Dimensões e características da Web brasileira: um estudo do .gov.br" apenas 2% das páginas Web governamentais brasileiras são acessíveis, aderentes ao e-MAG – Modelo de Acessibilidade de Governo Eletrônico que é orientado por padrões internacionais como a WCAG [2] [3].

Em um contexto que reúne a Educação e a Web, os Ambientes Virtuais/Digitais de Aprendizagem contribuem como suporte na aprendizagem oferecendo ferramentas que auxiliam na construção do conhecimento, produção coletiva e relacionamentos interpessoais. Dentro dos desafios de incluir sem excluir, a Tecnologia Assistiva (TA) fornece ferramentas que apoiam PcD a desempenharem suas atividades escolares e profissionais. Considerando que a educação é uma das iniciativas que mais pode proporcionar resultados positivos em relação à inclusão, os ambientes digitais despontam como espaços compartilhados de aprendizagem. No entanto, para que esses ambientes possibilitem criar espaços ricos de construção e de convivência, é necessário que sejam aperfeiçoados e possam garantir a autonomia e o acesso a todas as pessoas, sem nenhum tipo de obstáculo a estes ou dentro destes ambientes.

O objetivo do presente trabalho foi identificar um editor de texto acessível Web conforme a recomendação WCAG, realizar testes de usabilidade de validação, desenvolver novas funcionalidades para o editor e averiguar os recursos desenvolvidos com novos testes de usabilidade. O editor de texto escolhido foi o CKEditor. Dentre os critérios para eleger o editor, considerou-se: permitir a integração com outros ambientes Web, manter os princípios do software livre e utilizar as mesmas tecnologias e recursos computacionais do ambiente Eduquito, ambiente digital/virtual de aprendizagem idealizado pelo Núcleo de Informática na Educação Especial (NIEE) da Universidade Federal do Rio Grande do Sul (UFRGS), projetado para operar como ambiente de inclusão sócio digital, oferece recursos de acessibilidade e tecnologia assistiva para pessoas com deficiência auditiva, visual, pessoas não alfabetizadas, entre outras [4] [5].

Os requisitos implementados para o editor de texto, de forma a suprir as necessidades dos usuários que possuem deficiência visual, foram levantados em: entrevistas com a equipe do NIEE, testes com usuário cego, conhecimento apropriado na pesquisa

sobre acessibilidade e nos documentos da WCAG. Estes requisitos foram validados através de testes de usabilidade com usuários cegos.

O presente artigo está organizado da seguinte forma: a seção 2 explora o assunto usabilidade fornecendo características para uma boa *interface* com usuário e algumas regras que podem ser aplicadas em design universal. A seção 3 introduz o assunto editores de texto acessíveis para Web, faz uma comparação e apresenta os passos percorridos para a identificação, especificação e implementação de melhorias para o CKEditor. Na seção 4 são apresentados os testes realizados com usuários cegos, para averiguação das melhorias desenvolvidas para o CKEditor. Por fim, as considerações e trabalhos futuros.

2. 2. ACESSIBILIDADE E DESIGN UNIVERSAL

Segundo Nielsen [6], a facilidade de aprendizado, a eficiência de uso, a facilidade de memorização, a baixa taxa de erros e a satisfação subjetiva são os cinco atributos da usabilidade. Conforme o autor, quando usuários inexperientes conseguem atingir resultados desejados em um curto espaço de tempo, então um sistema é dito como de fácil aprendizado. O autor também descreve alguns princípios como a previsibilidade, a familiaridade e o poder de generalização como sendo conceitos relacionados à facilidade de aprendizado. A previsibilidade é a característica em que o usuário, através do histórico de experiências passadas, consegue determinar resultados para uma interação futura no mesmo sistema. A familiaridade refere-se aos modelos mentais construídos na utilização de experiências anteriores em outros sistemas ou no mundo real, para determinar ações no aprendizado de um sistema. A generalização é a capacidade de o usuário estender conhecimentos adquiridos em um sistema para outros sistemas análogos [6].

De acordo com [7], quando se trata da Web, uma página deve reunir cinco boas práticas para certificar que os usuários possam compreendê-la o máximo possível:

1. Hierarquia clara: manter a hierarquia para cada página de Internet, com a clareza e a precisão do relacionamento entre os elementos.
2. Convenções: utilizar convenções para que o usuário não precise pensar para descobrir onde está o elemento que procura, pois já possuirá o modelo mental da convenção.
3. Áreas claramente definidas: permitir aos usuários decidir, de forma rápida, quais áreas da página focar e quais podem ignorar sem perdas.
4. Deixar óbvio o que pode ser alterado: a *interface* pode auxiliar os usuários tornando óbvio o que eles podem fazer, guiando o usuário a utilizar os elementos de interesse, como botões de pesquisa, *links* sobre imagem, entre outros.
5. Manter a confusão no menor nível possível: para [7] há dois tipos de confusão, uma ocorre quando há excesso de coisas que o usuário pode realizar, quando tudo na página chama muito atenção; a outra é a confusão de segundo plano, quando a página possui elementos que causam distrações ao usuário.

Quando se fala sobre *design* universal (termo adotado no desenvolvimento de diferentes produtos, que tem origem na área da Arquitetura para conceituar obras que oferecem os mesmos benefícios a todas as pessoas que a utilizam) amplia-se as

possibilidade de usabilidade. Conforme [8], o *design* universal pode ser caracterizado como: o cuidado no processo de criação de produtos, no sentido de que esses produtos sejam comercialmente acessíveis a indivíduos portadores das mais variadas competências, em diferentes e múltiplas situações cotidianas. Entretanto, levando em consideração a enorme possibilidade de situações do dia-a-dia e o vasto campo de capacidades dos indivíduos, o *design* universal preocupa-se em proporcionar o acesso ao produto ao maior número possível de pessoas. De forma geral é possível afirmar que são duas as preocupações que recebem enfoque no *design* universal:

1. A flexibilidade no desenvolvimento de produtos, no sentido de que possam ser utilizados sem a necessidade de modificações ou do uso de tecnologias assistivas.
2. O desenvolvimento de produtos adaptáveis a tecnologias assistivas, visando seu uso por aquelas pessoas que não possuem a possibilidade de acesso e uso desses produtos de outra forma.

Uma das exigências do *design* universal no desenvolvimento de um produto é que todas as informações sejam percebidas mesmo sem a visão, audição, habilidade normal de leitura e aprendizado, percepção de cores, e sem causar distúrbios mentais [8]. Assim, enquanto o *design* universal preocupa-se com a universalização de uso dos produtos comercializados, o *design* acessível preocupa-se também com a adaptação desses produtos para que possam ser utilizados e auxiliem aqueles indivíduos que apresentam alguma limitação. Seria, na realidade, uma troca de benefícios entre os dois: enquanto pessoas com deficiência podem se beneficiar das características do *design* universal, os demais indivíduos também podem beneficiar-se das características do *design* acessível como, por exemplo, utilizando as rampas de acesso para a passagem de carrinhos de bebês e de compras.

Um *software* pode ser considerado acessível a uma pessoa, se essa, independentemente de suas limitações, conseguir utilizá-lo, realizar as mesmas funções e atingir os mesmos resultados que outras pessoas sem necessidades especiais, com conhecimentos e treinamento semelhantes. Desta maneira, a acessibilidade determina a competência de um sujeito acessar e usar um produto de modo eficaz e eficiente, atingindo resultados equivalentes por meios diversos.

O *design* acessível encontra na iniciativa WAI o apoio para tornar a Web mais acessível para pessoas com deficiência, incluindo cegueira e baixa visão, surdez e baixa audição, dificuldades de aprendizagem, limitações cognitivas, limitações de movimentos, incapacidade de fala, fotossensibilidade e suas combinações. A WAI promove a padronização de melhorias de acessibilidade através de recomendações reunidas em três diretrizes [9]:

1. Web Content Accessibility Guidelines (WCAG): designada aos desenvolvedores de páginas Web que queiram aplicar em seus *sites* recursos de acessibilidade.
2. User Agent Accessibility Guidelines (UAAG): destinada aos desenvolvedores de Web Browsers Players de Mídia, além de apresentar perspectivas das tecnologias assistivas.
3. Authoring Tool Accessibility Guidelines (ATAG): orienta fabricantes de Sistema de Gerenciamento de Conteúdo (CMS) a implementarem funcionalidades que auxiliam o desenvolvedor de conteúdo a seguir as recomendações especificadas na WCGA.

Essas diretrizes descrevem as características necessárias para o alcance de diferentes níveis de acessibilidade e, incluem listas de referência e técnicas de implementação. Também desenvolvem técnicas de melhorias das ferramentas de avaliação e reparação de *sites* através do trabalho da Evaluation and Repair Tools Working Group (GTERT) [9].

Como forma de atender às diversas necessidades do universo de pessoas e organizações que utilizam o WCAG, a WAI disponibiliza vários níveis de abordagem. A estrutura deste documento está organizada da seguinte forma [9]:

- Princípios: perceptível, operável, compreensível e robusto. Estes princípios formam a base para que o acesso a conteúdos da Web por todas as pessoas seja possível.
- Recomendações: para cada um dos princípios há um conjunto de recomendações. São doze recomendações para todos os princípios que, se forem seguidas, tornam o conteúdo mais acessível aos usuários com diferentes incapacidades.
- Critério de sucesso: servem como métricas para medir o nível de conformidade com a WCAG 2.0. Os critérios podem ser testados através de ferramentas e da avaliação humana. Com isso, é possível determinar se um critério de sucesso foi cumprido e com qual grau de confiança. Para diferenciar os níveis de conformidade com estes critérios, foram criados os níveis de classificação: A (o menos elevado), AA e AAA (o mais elevado). A W3C criou logos que exprimem esta classificação para serem adicionados aos *sites*.
- Técnicas de tipo Suficiente e de tipo Aconselhada: para cada uma das recomendações e critérios de sucesso existem técnicas de carácter informativo. A primeira é dita suficiente por garantir o cumprimento dos critérios de sucesso e a segunda vai além do que é requerido para o cumprimento dos critérios de sucesso.

O projetista de sistemas Web que seguir as recomendações disponibilizadas no WCAG 2.0 promoverá a inclusão de mais pessoas, aumentando o alcance do seu produto. Este trabalho segue as recomendações na construção de recursos para um editor de texto Web, já que o intuito é possibilitar que o editor possa ser integrado a Ambientes Virtuais/Digitais de Aprendizagem, que é uma ferramenta da Web. Os recursos de acessibilidade que foram utilizados para implementar as melhorias descritas na seção 3, seguiram as recomendações do documento WCAG 2.0.

3. 3. EDITOR DE TEXTOS ACESSÍVEL

Um dos objetivos deste trabalho foi escolher um editor de texto acessível para Web e aplicar melhorias de acessibilidade e usabilidade, ou seja, preocupar-se com seu *design* universal. Foi feita uma análise comparativa entre os editores CKEditor 4 e TinyMCE 3.x e, com base nesta análise, o editor de texto para Web CKEditor foi escolhido por contemplar todas as características necessárias para este trabalho, como:

- Compatibilidade com as tecnologias JavaScript, Java e PHP que são empregadas no desenvolvimento do Eduquito.
- Corretor ortográfico para Português do Brasil.
- Compatibilidade com os navegadores: Mozilla Firefox, Internet Explorer, Opera, Safari e Chrome.
- Conformidade com as recomendações de acessibilidade da W3C.
- Acesso aos recursos gráficos por teclado.

- Aplicação WYSIWYG.
- Open Source.
- Documentação do código fonte e manuais de apoio para desenvolvedores.
- Compatibilidade com leitores de tela.
- Sem dependências de tecnologia.

A versão avaliada foi a CKEditor 4.1.2 [10]. Trata-se de um editor de texto para Web WYSIWYG que disponibiliza os recursos mais populares já consolidados nos editores para *desktop* e pode ser facilmente incorporado a uma página Web. Uma das principais características do editor é a compatibilidade com tecnologias de servidores e navegadores como: Internet Explorer, Mozilla Firefox, Google Chrome, Safari e Opera. O CKEditor é um projeto *open source* distribuído sob as licenças General Public License (GPL), Lesser General Public License (LGPL) e Mozilla Public License respectivamente (MPL).

O CKEditor (Figura 1) contempla padrões de recomendações de acessibilidade da WCAG1.0 e WCAG2.0. O editor é compatível com o leitor de tela Jaws versão 14, homologada com o navegador Firefox versão 21, possui compatibilidade com o recurso Thema Alto Contrast dos navegadores Firefox versão 21 e Internet Explorer versão 9. Para utilização com dispositivos móvel ele é compatível com o iPhone e iPad, utilizando o sistema operacional iOS6, embora a comunidade reconheça que a usabilidade precisa ser melhorada [10].

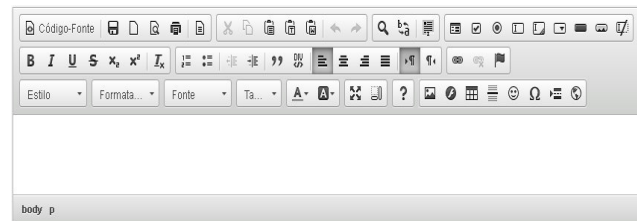


Figura 1. Figura 1. Editor de Texto CKEditor versão 4.1.2 com todas ferramentas.

A seguir, apresenta-se os passos percorridos para a identificação, especificação e implementação das melhorias para o CKEditor.

3.1 3.1 Identificação

Para o levantamento dos requisitos de melhoria foram utilizadas quatro abordagens:

- (1) realização de testes de usabilidade com usuário com boa visão utilizando venda nos olhos,
- (2) realização de teste de usabilidade com usuário cego,
- (3) entrevista com o NIEE,
- (4) recomendações do W3C (descrito na seção 2).

No teste de usabilidade com usuário sem cegueira utilizando venda nos olhos (1) o objetivo foi compreender as barreiras enfrentadas por pessoas cegas ao utilizarem um editor de texto, quais recursos que favorecem a acessibilidade e usabilidade na interação com o computador. Para realização do teste, o usuário, autor deste trabalho, teve seus olhos vendados e realizou algumas tarefas como: acessar a área de edição do editor de texto, encontrar informações na caixa de diálogo de ajuda de

acessibilidade, utilizar algumas ferramentas como, negrito e itálico, entre outras atividades. O aprendizado com esta experiência auxiliou no entendimento dos recursos de acessibilidade, bem como, na elaboração dos testes com usuários cegos e na condução da entrevista com o grupo de pesquisa.

A entrevista com o NIEE (3) -grupo de pesquisa que desenvolve o Eduquito- teve como finalidade especificar as características e funcionalidades que contemplassem as necessidades do ambiente digital de aprendizagem. O NIEE tem um sólido trabalho e produz conhecimentos na área de acessibilidade, o que foi muito útil para o levantamento e confirmação de requisitos de melhoria para o editor de texto. Foram realizadas duas reuniões: a primeira com os pesquisadores e a segunda com a equipe técnica de desenvolvimento do Eduquito. Na primeira reunião, os requisitos levantados nos testes com o usuário foram validados com os pesquisadores. Na conversa com a equipe técnica, foram identificadas as possibilidades de aplicação do editor acessível ao Eduquito e validadas as características técnicas de implementação.

Na entrevista, reforçando o que já havia sido definido como prioridade para a implementação, foi solicitado que o editor atendesse às recomendações de acessibilidade contidas no documento WCAG 2.0. Outro requisito exigido na entrevista foi de que as melhorias implementadas deveriam ser disponibilizadas em forma de *plugin*, ficando independente do código-fonte original, e que deveriam ser disponibilizadas para a comunidade que desenvolve o CKEditor.

O teste com o usuário cego (2) resultou nos novos requisitos implementados para o CKEditor. Para o planejamento e aplicação do teste, seguiram-se as recomendações de autores da área de usabilidade como [6], [7] e [8]. O método contempla os seguintes passos:

1. Plano de teste e *script*: trata da organização e planejamento do teste, assim é elaborado o plano de teste (que contempla: objetivo, local, duração, suporte, recursos, perfil do usuário, agendamento do teste, entre outros itens) e o *script* (roteiro para a condução do teste). O *script* inclui etapas como: (a) apresentação, (b) ambientação inicial, (c) tarefas e (d) perguntas de encerramento [7].
2. Teste de Compreensão: o editor de texto é apresentado para o usuário e é analisado se o mesmo consegue compreender a proposição do editor, sua organização e funcionamento.
3. Teste de Tarefas-chave: o usuário realiza tarefas predefinidas. A realização da tarefa evidencia os problemas de usabilidade e acessibilidade. Neste momento o usuário também contribui com o levantando de requisitos e sugestões.
4. Classificação dos problemas encontrados: ruído (diminuição do desempenho na realização de uma tarefa), obstáculo (quando ao realizar uma ação, o usuário faz várias tentativas, mas consegue aprender e executar a tarefa) e barreira (quando na interação com o sistema, o usuário esbarra no problema sucessivas vezes e não aprende a suplantá-lo) [8].

Na aplicação das estratégias metodológicas procurou-se seguir as orientações defendidas por [11] e [12], sobretudo naquilo que concerne à relação entre aplicador do teste e o participante. Entende-se que a simetria é condição básica para a condução de qualquer estudo, já que a fabricação de fatos é sempre resultado de um agenciamento entre humanos e não humanos. Ser simétrico em pesquisas com participantes humanos, entre outras coisas, implica em considerar a legitimidade da participação do sujeito como coautor das considerações realizadas.

Os relatos a seguir descrevem as atividades executadas com o usuário durante os experimentos com o editor. Tais relatos têm base nos registros das anotações, dos vídeos e das respostas coletadas durante as atividades, bem como nos comentários e sugestões feitos pelo usuário durante o teste. Desta forma, documentou-se todo o processo mantendo foco principal na interação, no comportamento do usuário e em seus comentários, para poder claramente identificar os pontos de fragilidade do editor. O teste foi aplicado com um usuário cego, o funcionário Cristian Evandro S., que trabalha no Núcleo de Ação Comunitária da Universidade de Santa Cruz do Sul. O usuário usa o computador no trabalho e em casa e com frequência navega na Internet.

Para realização do teste foi reservado uma data e horário combinado com o usuário. Para facilitar a aplicação, utilizou-se o espaço onde o usuário trabalha e o computador que ele opera diariamente. Essa abordagem foi importante para que o usuário ficasse à vontade e o teste transcorresse de maneira natural. Como suporte computacional para realização do teste foi utilizado um computador com caixas de som (que possibilitasse escutar o leitor de tela), a gravação do áudio reproduzido pelo leitor, gravação das conversas com o usuário e anotações da interação do usuário com o editor de texto. O leitor de telas utilizado foi o NVDA versão 2013.1.1, para capturar o áudio do microfone e capturar a tela do computador foi usado o *software* CamStudio versão 2.7, e o navegador era o Mozilla Firefox ESR 17.0.06. Todos estes *softwares* foram instalados no sistema operacional Microsoft Windows 7 versão ultimate 64 bits.

Seguindo o *script* planejado, a sessão de teste iniciou com a etapa de apresentação (a) sobre o propósito do mesmo. Esse momento teve duração de 5 minutos. Na etapa de ambientação inicial (b) o usuário foi instigado a descobrir funcionalidades e recursos do editor, ele falava em voz alta tudo que estava pensando e justificava todas as suas ações. Neste momento, foram feitas algumas perguntas referentes à impressão que o usuário teve diante do primeiro contato com o editor. Este momento teve duração de 10 minutos.

Ainda na etapa de ambientação, a avaliação inicial teve como objetivo descobrir se o editor de texto fornecia recursos suficientes para o usuário constatar que o conteúdo da página apresentada a ele se tratava de um editor de texto, além de possibilitar ao usuário encontrar a área de edição do editor. O usuário descobriu que se tratava de um editor de texto, mas teve dificuldades em encontrar a área de edição. Neste momento foi constatado um problema do tipo obstáculo, pois quando o foco estava na área de edição, houve um excesso de mensagens faladas pelo NVDA, causando confusão. A partir da terceira tentativa navegando com a tecla *tab*, o usuário acessou a área de edição e aprendeu do que se tratava. A sugestão de melhoria para este problema foi investigar se o excesso de mensagens resultou do texto fornecido pelo editor para o NVDA ou houve interferência do navegador. Como resultado da investigação, observou-se que o problema era do navegador. Também foi feita a comparação das mensagens em diferentes navegadores. Conforme o resultado da análise, o código do editor foi alterado para que, ao receber o foco do navegador, envie a mensagem para o NVDA de “Aplicação, editor de texto, área de edição”.

Ainda na fase de ambientação, foi solicitado ao usuário que escolhesse uma tarefa usual para executá-la no editor. Nesse momento o usuário, além de executar a tarefa escolhida por ele,

realizou, por curiosidade, diversas tarefas que estavam previstas para a terceira etapa. Segundo [7], sempre que for possível, deve-se deixar o usuário dar opinião na escolha da tarefa. Seguindo esta recomendação, o usuário não foi interrompido e as tarefas que seriam avaliadas na terceira etapa foram observadas nesta fase de ambientação, de modo a obter o máximo de naturalidade do usuário.

O usuário escolheu colocar uma palavra em negrito e, para descobrir a tecla de atalho correspondente, tentou acessar a ajuda pressionando as teclas *Alt* + zero, conforme o aviso sonoro lido pelo NVDA quando o editor recebeu foco. Assim, ele pressionou a tecla *Alt* e a tecla zero do teclado numérico e a ajuda não foi exibida, então pressionou a tecla *Alt* e a tecla zero do teclado alfanumérico e obteve sucesso. Nesse momento foi detectado o problema do tipo ruído, caracterizado pela tentativa do usuário de utilizar a tecla zero do teclado numérico sem sucesso. A sugestão de melhoria foi possibilitar a utilização da tecla zero do teclado numérico, por ser mais facilmente encontrada, como foi constatado ao observar o usuário contar as teclas do teclado alfanumérico a partir da tecla um até a tecla zero. O usuário relatou que tem dificuldades de encontrar a tecla zero no teclado alfanumérico, pois não há nenhuma marcação no teclado que auxilie o usuário cego a encontrá-la.

Após conseguir acessar a ajuda para acessibilidade do editor de texto, o usuário tentou encontrar o grupo da ajuda referente às teclas de atalho, mas não obteve sucesso, visto que possuía apenas a possibilidade de pular as linhas da ajuda com a tecla direcional para baixo e, como havia muitas linhas na ajuda, ele não conseguiu encontrar. A elevada quantidade de informações em um mesmo local se configurou em uma barreira, o usuário deixava o NVDA lendo a tela por um longo tempo. Como sugestão para organizar a ajuda, facilitando a procura pelos diversos assuntos, a tecla *tab* pode ser utilizada para percorrer grupos da ajuda.

Na terceira etapa, o *script* previa a realização de tarefas (c). Esse momento teve duração de 20 minutos. Nessa etapa foram detectados problemas do tipo barreira, quando o usuário tentou alterar o tamanho da fonte e descobrir se a ferramenta negrito estava pressionada. As sugestões de melhoria foram: colocar a ferramenta “tamanho da fonte” como primeiro item do grupo e tornar mais clara a mensagem de ferramentas como o “negrito” quando recebe o foco.

Para finalizar o teste algumas perguntas de encerramento (d) foram feitas ao usuário, como: sugestões de melhorias, comentários sobre o teste ou dificuldades encontradas. Esse momento teve duração de 5 minutos e foram feitos alguns questionamentos no intuito de validar problemas percebidos durante o teste e validar requisitos identificados em etapas anteriores. Algumas perguntas feitas foram:

“Auxiliaria no processo de descoberta das teclas de atalho se estivessem descritas no texto disponibilizado ao NVDA quando a ferramenta recebe o foco na barra de ferramentas? (...). Visando diminuir o percurso realizado para descobrir a tecla de atalho de uma ferramenta e o percurso realizado para ativar ferramentas na barra de ferramentas, seria uma boa abordagem criar um *pop-up*, acessível via tecla de atalho, que serviria para consulta de teclas de atalho e seleção de ferramenta? (...)”

Nessa etapa de perguntas de encerramento importantes melhorias foram definidas. O usuário observou que havia poucas teclas de

atalho e sugeriu que fossem adicionadas para todas as ferramentas. Outras dificuldades e sugestões relatadas pelo usuário foram observadas durante as fases de ambientação e tarefas. O usuário aprovou a sugestão de criar o *pop-up* para pesquisa de teclas de atalho e sugeriu que mais ferramentas do editor de texto sejam contempladas com teclas de atalho. O *pop-up* para pesquisa de teclas de atalho foi idealizada após estudos acerca de acessibilidade e usabilidade. Este *pop-up* permitirá que o usuário acesse a tecla de atalho de uma ferramenta. Outra sugestão do usuário foi permitir que editor disponibilizasse ao NVDA a tecla de atalho da ferramenta que recebe o foco.

O relato do teste apresentado aqui é uma síntese das ocorrências do uso em cada etapa de teste, classificação dos problemas encontrados e sugestões de melhorias. O registro detalhado da sessão de teste com o usuário para identificação de requisitos para a melhoria do editor de texto pode ser encontrado em [13].

3.2 3.2. Especificação

Com base na análise e nos dados obtidos na fase de identificação dos requisitos para a melhoria do editor de texto, de forma a suprir as necessidades dos usuários que possuem deficiência visual, foram definidos as novas funcionalidades para o editor, são elas:

1. Melhorar a mensagem quando o editor de texto recebe foco.
2. Tornar mais fácil a pesquisa e leitura da “Ajuda Para Acessibilidade”.
3. Possibilitar a utilização da tecla zero do teclado numérico para acessar a ajuda utilizando tecla de atalho.
4. Disponibilizar a descrição da tecla de atalho quando o botão receber foco na barra de ferramentas.
5. Desenvolver caixa de diálogo com a lista de comando e suas teclas de atalho.
6. Contemplar mais comandos com teclas de atalho.
7. Reorganizar a barra de ferramentas.

3.3 3.3. Implementação

Os requisitos foram implementados de forma a resolver os problemas de acessibilidade e usabilidade detectados na etapa de identificação. A compatibilidade do CKEditor em diferentes navegadores, como descrito anteriormente, é referente a sua execução. Já a compatibilidade dos recursos de acessibilidade com navegadores e tecnologias assistivas são mais limitadas. Segundo o *site* do editor, os recursos de acessibilidade do CKEditor são compatíveis com o navegador Mozilla Firefox e o leitor de tela Jaws. Sendo assim, os recursos de acessibilidade implementados foram testados no Firefox. O desenvolvimento seguiu as recomendações do guia do desenvolvedor CKEditor, para que as modificações fossem disponibilizadas para a comunidade que mantém o editor de texto.

Todas as ferramentas e funcionalidades foram criadas e organizadas em forma de *plugins*, o que facilita adicionar e remover recursos conforme a necessidade do desenvolvedor (Figura 2). A internacionalização está implementada em todos os *plugins* sendo uma das exigências para a comunidade aceitar *plugins* de quem queira contribuir.

Outro conceito utilizado é de *dialogs* (caixas de diálogos), ou seja, os *plugins* podem cadastrar caixas de diálogo que são pré-definidas. O código que compõem os comandos, ao serem executados, podem abrir estas caixas de diálogos. Para isso, o editor provê ferramentas que facilitam a criação de *plugins*

tornando a interface padronizada, o código reutilizável e de fácil entendimento.

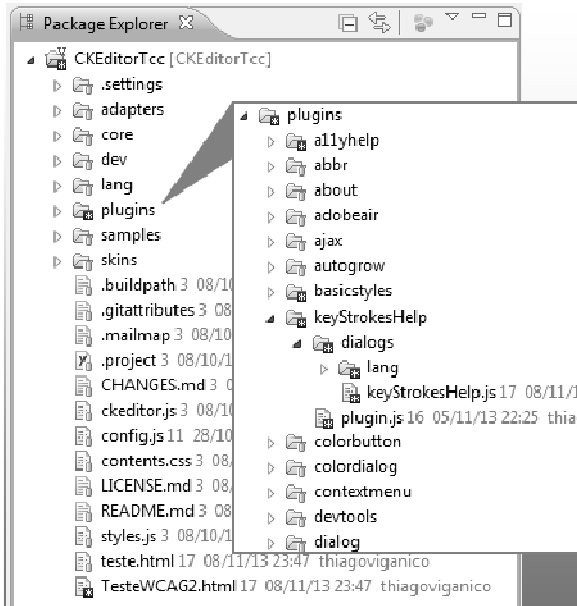


Figura 2. Código fonte do CKEditor.

A seguir faz-se um breve relato dos requisitos implementados.

1. Melhorar a mensagem quando o editor de texto recebe foco. Foi realizada uma comparação das mensagens disponibilizadas para o NVDA nos navegadores Firefox, Chrome e Internet Explorer. A Tabela 1 mostra esta comparação. Concluiu-se que o excesso de mensagens não é um problema no editor de texto. O navegador Chrome forneceu para o NVDA as informações corretas sem repetir palavras. Assim, esta melhoria não resultou em um requisito implementável, mas em uma investigação que contribuiu para a descoberta de um fato e a necessidade de melhorias na acessibilidade de navegadores Web.

Tabela 1. Comparação das mensagens.

| Navegador | Mensagem lida pelo NVDA |
|-------------------|--|
| Firefox | editor de texto editor 1 editor de texto editor 1 frame internal pressione Alt+0 para ajuda editor de texto editor 1 Documento editável |
| Chrome | editor de texto editor 1 Documento editável Sessão editável em branco |
| Internet Explorer | editor de texto editor 1 editor de texto editor 1 |

2. Tornar mais fácil a pesquisa e leitura da “Ajuda Para Acessibilidade”. Os grupos “Geral” e “Comandos” foram removidos já que causavam confusão e não agregavam informações relevantes ao usuário. Para eliminar o problema

que foi classificado como barreira, a navegabilidade pelo conteúdo da ajuda foi alterada. Foram utilizadas as marcações de lista, item e descrição do item para exibir o conteúdo da ajuda. Desta maneira o usuário pode utilizar a tecla I para percorrer os itens da lista, saber a quantidade de itens e quando chegou ao final da lista. A lista de teclas de atalho foi removida, já que estava programada de forma estática, ou seja, não era atualizada caso alguma tecla de atalho fosse atribuída para alguma ferramenta ou alterada. Para a descrição das teclas de atalho na ajuda, adicionou-se um novo item que explica a existência da caixa de diálogo com a lista de teclas de atalho e como acessá-la. A Figura 3 mostra como a caixa de diálogo para ajuda de acessibilidade ficou após alteração.

A mensagem disponibilizada ao NVDA quando a caixa de diálogo é aberta foi alterada da seguinte forma: “Conteúdos da Ajuda. Pressione ESC para fechar esta janela. Para navegar pelos grupos da ajuda, pressione L, para navegar pelos itens dos grupos, pressione I, para navegar linha a linha. Para voltar os grupos e seus itens, pressione a tecla SHIFT junto com as teclas I ou L. Você pode navegar livremente utilizando as setas para cima e para baixo”. Também foi alterada a ordem dos itens de modo que os primeiros sejam os mais relevantes para o usuário. Entende-se que utilizar a barra de ferramentas e saber como encontrar a descrição das teclas de atalho é tarefa básica para que o usuário possa utilizar o editor de texto, por isso são os primeiros itens da lista.

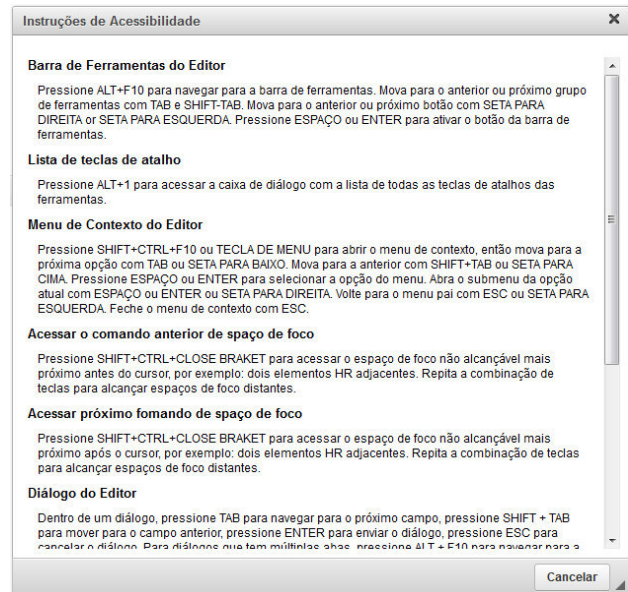


Figura 3. Novo item Lista de teclas de atalho na caixa de diálogo Instruções de Acessibilidade.

3. Possibilitar a utilização da tecla zero do teclado numérico para acessar a ajuda utilizando tecla de atalho. Para atribuir teclas de atalho a algum comando, o editor possui um método que deve ser utilizado por programadores que queiram atribuir uma tecla de atalho o comando do seu plugin. Desta forma, a lista de teclas de atalho fica centralizada em um mesmo ponto do código favorecendo o reuso de código. Assim, foi possível buscar a listas de teclas

de atalho em várias implementações, de forma simples e atualizada. A Figura 4 mostra o código utilizado para configurar a tecla zero do teclado numérico para abrir a ajuda de acessibilidade. O primeiro parâmetro é o código ASCII¹ da tecla *Alt*, definida pela constante CKEDITOR.ALT, o segundo parâmetro é o código ASCII da tecla ZERO do teclado numérico, e por fim, o nome do comando que receberá a atribuição da tecla de atalho, que no caso é `al1yHelp` que corresponda a ajuda de acessibilidade.

```
editor.setKeystroke( CKEDITOR.ALT + 96 /*0*/, 'al1yHelp' );
```

Figura 4. Atribuindo uma tecla de atalho para um comando.

- Disponibilizar a descrição da tecla de atalho quando o botão receber foco na barra de ferramentas. Para aumentar a possibilidade do usuário descobrir as teclas de atalho das ferramentas utilizando o menor percurso possível, foi adicionada a descrição da tecla de atalho na mensagem disponibilizada ao leitor de tela quando as ferramentas recebem foco. Quando o usuário acessa a barra de ferramenta é necessário que o editor forneça informações que possibilite ao usuário cego saber que está com o foco sobre um botão, qual o tipo de botão, o nome da ferramenta e o estado do botão (pressionado ou não pressionado).

A WAI define diretrizes que possibilita fornecer todas estas informações ao usuário, desta forma recursos WAI-ARIA foram utilizados. Este requisito descreve que a tecla de atalho deve ser disponibilizada para o leitor de tela quando os botões da barra de ferramentas receberem foco. Para tal, foi alterado no código-fonte o atributo que armazena as teclas de atalho e foi adicionada a descrição na mensagem que é lida quando o botão recebe foco. Como resultado, na execução do editor, o código *javascript* verifica o atributo que define o estado do botão.

- Desenvolver caixa de diálogo com a lista de comando e suas teclas de atalho. A listagem das teclas de atalho foi implementada para estar sempre atualizada. Assim, serão exibidas apenas as teclas de atalho cuja ferramenta correspondente esteja disponível no editor de texto. Fornecer a tecla de atalho de uma ferramenta que não esteja disponível no editor causará confusão ao usuário cego, já que não saberá se a ferramenta realmente está disponível. A Figura 5 apresenta como ficou a caixa de diálogo com a listagem das teclas de atalho. Inicialmente, a informação de teclas de atalho era a seguinte: “comando negrito, pressione *Ctrl+B*”. Entretanto, conforme sugestão dos usuários nos teste de validação das melhorias, foi retirada a formação do texto e deixada apenas a mensagem “Negrito, *Ctrl+B*”. Para alertar o usuário da existência deste recurso, foi adicionado junto à mensagem, que é disponibilizada ao leitor de tela quando o editor de texto recebe foco, a frase “Pressione *Alt+I* para lista de teclas de atalho”. Também foi adicionado um item na “Ajuda de acessibilidade” que descreve este recurso e como acessá-lo.

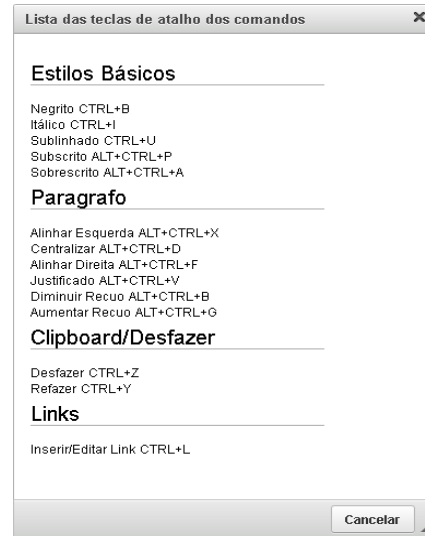


Figura 2. Figura 5. Caixa de diálogo com a lista de teclas de atalho.

- Contemplar mais comandos com teclas de atalho. Os comandos que possuem teclas de atalho são: desfazer, refazer, negrito, itálico, sublinhado e *link*. Foi atribuídas teclas de atalho para os comandos subscripto, sobrescrito, alinhar a esquerda, centralizar, alinhar a direita, justificado, diminuir recuo, aumentar recuo. Para escolha das teclas de atalho, foram analisadas as teclas de atalho dos navegadores Firefox, Chrome, Internet Explorer e Safari de modo que as novas teclas de atalho não conflitassem com teclas de atalho destes navegadores. A Figura 6 mostra a lista de todos os comandos que receberam teclas de atalho.

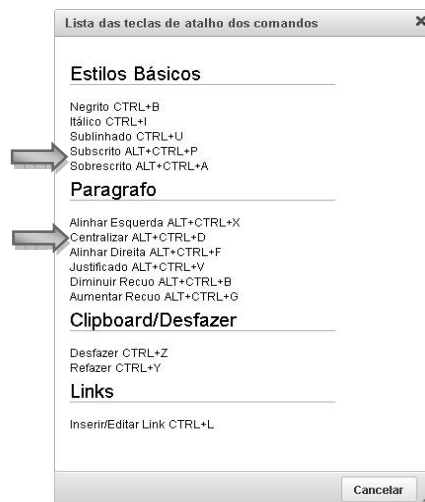


Figura 3. Figura 6. Lista de teclas de atalho.

- Reorganizar a barra de ferramentas. Usuários cegos utilizam, em sua maioria, o teclado para interagirem com a interface gráfica. Nesta lógica, o caminho mais rápido para alcançar uma ferramenta não é o caminho percorrido pela seta do mouse e sim a quantidade de teclas pressionadas para alcançar a ferramenta. A Figura 7 mostra a nova organização da barra de ferramentas.

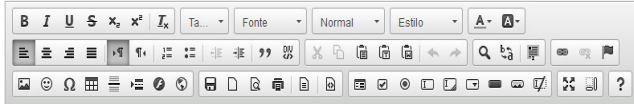


Figura 4. Figura 7. Barra de ferramentas do CKEditor com nova ordenação.

4. 4. TESTES COM USUÁRIOS CEGOS

Esta seção descreve a análise dos testes realizados com objetivo de constar se as melhorias desenvolvidas eliminaram os problemas de usabilidade e acessibilidade encontrados no teste de avaliação. Para averiguar as melhorias implementadas, o teste elaborado segue as mesmas recomendações do teste aplicado na etapa de identificação.

Os recursos computacionais utilizados para aplicar o teste foram: um *notebook* e um teclado de computador *desktop* (teclados de notebooks costumam ter padrões de teclas diferentes que pode dificultar a utilização pelo usuário). O leitor de tela utilizado foi o NVDA versão 2013.1.1, para capturar o áudio do microfone e capturar a tela do computador foi usado o *software* Camtasia versão 8, e o navegador utilizado o Mozilla Firefox ESR 17.0.06. Todos os softwares foram instalados no sistema operacional Microsoft Windows 7 versão ultimate 64 bits.

Os usuários que auxiliaram nos testes foram Cristian Evandro S., Ricardo S. K. e Leonardo S. ambos cegos. Todos os usuários possuem cegueira total e um bom conhecimento em informática. Os testes foram aplicados individualmente, com base no roteiro e no script. O local utilizado para aplicar o teste com Cristian Evandro S. foi uma sala de reuniões localizada no setor onde trabalha. Os testes com Ricardo S. K. e Leonardo S. foram realizados no local que é destinado à utilização do computador nas suas residências.

4.1 Análise dos testes com usuários

Na fase de ambientação inicial analisou-se a capacidade dos usuários em descobrir elementos da *interface* gráfica do editor. Para isso, os usuários realizaram tarefas de sua escolha e interagiram livremente com o editor.

Após a ambientação inicial, iniciou-se a fase de tarefas, que consiste em solicitar aos usuários realizassem tarefas planejadas. Estas tarefas foram elaboradas para conduzir o usuário a utilizar os novos recursos implementados.

Por fim, chegou-se à fase de perguntas de encerramento, cujo objetivo foi questionar os usuários sobre as novas funcionalidades, bem como suas sugestões de melhorias.

Para descrever os testes são apresentadas a seguir considerações das perguntas do *script*.

- O usuário consegue encontrar a área de edição do editor de texto?

Sem qualquer explicação sobre como utilizar o editor de texto, os usuários foram instigados a encontrar a área de edição e utilizá-la livremente. Percebeu-se que o excesso de mensagens lidas quando o editor recebeu foco, causou confusão para um usuário, que precisou percorrer mais de uma vez o editor para analisar a mensagem lida quando recebia foco para ter certeza de estava na área de edição. Este excesso de mensagens não foi categorizado como um problema, visto que o usuário estava se familiarizando com o Firefox e NVDA (*softwares* utilizados para testar os recursos de acessibilidade do editor de texto) e nas atividades

seguintes o usuário facilmente localizou a área de edição. O excesso de mensagem ocorre porque o NVDA lê duas vezes o texto disponibilizado pelo editor de texto, ao receber foco. Esta repetição é um problema do navegador Firefox, como comentado anteriormente.

Uma observação importante foi que um recurso de acessibilidade auxiliou os usuários a perceberem que estavam na área de edição do editor de texto. Durante a mensagem lida pelo NVDA, quando o editor recebeu foco, a palavra “aplicação” chamou a atenção dos usuários que neste momento tiveram a certeza de estarem na área de edição. Esta palavra é lida pelo NVDA, pois o código HTML gerado pelo editor utiliza o atributo *role=“application”*, que é um recurso de acessibilidade definido pela WAI-ARIA.

Os leitores de tela detectam estas *roles* e adicionam às frases que são lidas quando os objetos da página recebem foco da descrição correspondente, que neste caso foi a palavra “aplicação”. As *roles* utilizadas de forma correta facilitam ao usuário cego saber que tipo de elemento da página está com foco (um botão, uma imagem, um aplicativo, entre outros). O objetivo deste item analisado foi descobrir se as mensagens disponibilizadas pelo editor de texto, ao receber foco, auxiliaram o usuário a encontrar a área de edição do editor de texto. A conclusão foi que os recursos de acessibilidade foram suficientes para que os usuários encontrassem a área de edição e aprendessem que pressionando as teclas *Ctrl+0* poderiam acessar a ajuda de acessibilidade.

- O usuário consegue acessar a barra de ferramentas do editor de texto?

Ainda na fase de ambientação, foi analisado se os usuários conseguiram acessar a barra de ferramentas. O usuário Cristian Evandro S., por ter participado do teste de validação do CKEditor, já sabia como acessar a barra de ferramentas. Já os demais usuários utilizaram a ajuda de acessibilidade e aprenderam que pressionando as teclas *Alt + F10* podiam acessar a barra de ferramentas. Outra observação importante é que o obstáculo detectado no teste de validação do CKEditor foi removido com a nova organização das ferramentas. O usuário localizou as ferramentas na primeira tentativa (que são os mais utilizados em um editor de texto), diferente do teste anterior, em que o usuário percorreu várias vezes a barra de ferramentas para encontrar os recursos exigidos para a realização das tarefas do teste. Nos três testes, os usuários utilizaram a tecla zero do teclado numérico para acessar a ajuda de acessibilidade.

- O usuário utilizou a caixa de diálogo com a lista de teclas de atalho para realizar as tarefas?

Na fase de tarefas, foi solicitado aos usuários que realizassem algumas tarefas com o objetivo de averiguar os benefícios da caixa de diálogo com a lista de teclas de atalho. Observou-se que dois usuários utilizaram a caixa de diálogo para descobrir as teclas de atalho e um usuário utilizou a descrição da tecla de atalho lida quando os botões das ferramentas recebiam foco. Um dos usuários, antes de utilizar algum comando, primeiro pesquisou a tecla de atalho para então utilizar o comando, esta observação demonstra a importância das teclas de atalho para pessoas cegas, como também, a utilidade deste recurso que foi desenvolvido.

- O usuário consegue utilizar a ajuda para acessibilidade?

Para esta análise foram solicitadas algumas atividades que exigissem ao usuário encontrar itens na ajuda, com isso foi possível verificar se os recursos de acessibilidade utilizados na

construção da caixa de diálogo da ajuda eram suficientes para que um usuário cego consiga utilizá-la.

O usuário Ricardo S. K. usou a ajuda para acessibilidade já na fase de ambientação para descobrir como acessar a barra de ferramentas. Esta análise mostra que os recursos de acessibilidade empregados na construção desta caixa de diálogo proveram meios de acessibilidade úteis para um usuário cego. Os outros usuários também utilizaram com sucesso a ajuda na realização das tarefas.

- Perguntas de encerramento

Na fase de perguntas de encerramento, que finaliza o teste, o usuário Cristian Evandro S. aprovou a nova organização da barra de ferramentas, pelo fato de ter diminuído o percurso até elas.

Todos os usuários se mostraram satisfeitos com os recursos para descobrir as teclas de atalho das ferramentas. Cristian Evandro S. sugeriu a possibilidade de executar o comando pressionando a tecla *enter* sobre o texto que descreve a tecla de atalho. Leonardo S. comentou que ao utilizar um *software* pela primeira vez, a primeira atividade que realiza é procurar a lista de teclas de atalho. Esta informação foi importante, pois comprovou que a melhoria implementada é útil e possui grande relevância de acessibilidade.

Outra sugestão de melhoria foi adicionar um áudio que descreva a *interface* gráfica do editor de texto. Segundo Cristian Evandro S., uma descrição auxiliaria o usuário cego a entender como a *interface* é apresentada. Esta descrição deve informar a posição que está a área de edição, a barra de ferramentas entre outros elementos gráficos do editor. Cristian Evandro S. e Leonardo S. sugeriram deixar apenas o nome da ferramenta e a correspondente tecla de atalho na caixa de diálogo da lista de teclas de atalho. A cada linha era lido “comando negrito, pressione *Ctrl+B*” sendo que a sugestão deles foi “negrito *Ctrl+B*”.

Outra sugestão foi retirar a formação do texto desta caixa de diálogo, pois o leitor de tela, quando configurado para isso, descreve o tipo de fonte, o tamanho da fonte, se está em negrito entre outros atributos de formação, causando confusão para o usuário, como demonstrado no teste com Leonardo S.

Na fase de perguntas de encerramento percebeu-se que os usuários não conheciam ou pouco conheciam os recursos WAI-ARIA e as recomendações de acessibilidades contidas no documento WCAG. Como foi o caso da caixa de diálogo com a lista de teclas de atalho, os usuários não sabiam que podiam acessar pressionando a tecla L, uma lista dentro de uma página da Internet. Questionados sobre outros recursos, como acessar o conteúdo principal de uma página de Internet (*role=main*), alguns responderam que desconheciam estes recursos.

4.2 4.2 Considerações dos testes

Os usuários tiveram algumas dificuldades na identificação da frase onde o cursor estava posicionado, qual a formatação do texto e outros recursos referentes à área de edição do texto.

Nenhumas destas dificuldades foram classificadas como problema de usabilidade ou acessibilidade do editor, pois a forma de ler o texto, a formatação do texto e o alerta ao usuário sobre as áreas de início do texto, de pontuação e outros alertas, são responsabilidades do leitor de tela.

Os testes realizados e as atividades definidas na *script* foram suficientes para concluir que todas as melhorias implementadas foram utilizadas e auxiliaram na acessibilidade e usabilidade do editor de texto Web. Conforme [7], três usuários são suficientes

para encontrar os problemas mais significativos. Na prática com os testes, também se observou que os pontos de problema identificados foram maiores com o primeiro usuário, no entanto foi importante realizar testes com mais de um usuário, para que fossem desconsideradas ações acidentais e confirmadas as ações usuais e úteis.

Os usuários sugeriram outras melhorias que foram implementadas após os testes de averiguação, como: alterar a descrição das teclas de atalho na caixa de diálogo com a lista de comandos e suas teclas de atalho. Com os testes também foi possível detectar que os problemas do tipo barreira, obstáculo e ruído, encontrados no primeiro teste de validação do CKEditor foram removidos.

5. 5. CONCLUSÕES E TRABALHOS FUTUROS

O indicador de páginas Web .gov.br aderentes às diretrizes de Acessibilidade do W3C em 2011 era de 4,82 % (que atendem a algum dos níveis de acessibilidade A, AA ou AAA) [14]. Considerando o tamanho da Web governamental brasileira (.gov.br) de 6.874.926 páginas e que 18,8% da população (35 milhões) de pessoas declararam, no Censo 2010, ter dificuldade de enxergar, mesmo com óculos ou lentes de contato [15], conclui-se que boa parte da população não consegue ou possui alguma dificuldade em acessar o conteúdo destas páginas. No Brasil, além de leis que garantam a acessibilidade, a inclusão e a qualidade de vida para todos, é preciso que a população se conscientize que aproximadamente 23,9% da população, ou seja, aqueles com pelo menos uma deficiência, seja visual, auditiva, motora ou mental, precisam ser inseridos no mercado de trabalho, nas salas de aula, nos eventos culturais e de lazer [15].

Os números apresentados pelo Censo evidenciam que a acessibilidade precisa, de fato, se tornar uma realidade no Brasil. Empresários devem perceber que promover acessibilidade em suas empresas, tanto em estabelecimentos físicos ou virtuais, não é um gasto sem retorno, mas a possibilidade de novas oportunidades. Possibilitar que 23,9% da população possa ter acesso a suas empresas é uma ótima maneira de aumentarem seus lucros e popularizar seu negócio.

Promover a acessibilidade, com certeza, é obrigação da sociedade. Entretanto, aprender a utilizar as TA's é tarefa das pessoas com deficiência. No decorrer da realização dos testes, percebeu-se que a pirâmide da inclusão só estará completa quando houver conhecimento sobre os recursos existentes para a promoção da acessibilidade e da usabilidade na Web para pessoas com deficiência utilizarem o computador.

A W3C promove a acessibilidade na Web através da Web Accessibility Initiative (WAI), desenvolvendo estratégias, diretrizes e recursos para auxiliar desenvolvedores de conteúdo Web a tornar suas páginas acessíveis. Esta iniciativa produz recomendações e recursos que, se forem seguidos, promovem acessibilidade para uma grande quantidade de pessoas com deficiência. Estas recomendações e recursos estão descritos no documento WCAG 2.0, que reúne uma gama de recomendações para tornar o conteúdo Web mais acessível. Outro importante documento é o WAI-ARIA que define informações semânticas, estruturas e comportamentos, a fim de permitir que tecnologias assistivas possam transmitir de forma adequada informações de conteúdos Web para pessoas com deficiência. O profissional que sabe utilizar as recomendações destes documentos poderá construir conteúdos Web fornecendo acessibilidade e garantindo

que as informações de suas páginas Web alcancem um maior número de pessoas.

Acreditando que a produção científica gerada pelas universidades pode contribuir para a resolução de problemas da sociedade e melhoria da sua qualidade de vida, este artigo disponibiliza uma investigação que reúne temas como acessibilidade e usabilidade na Web, ambientes virtuais de aprendizado e, principalmente, testes com usuário. Diante do desafio de disponibilizar um editor de texto acessível para Web, que possa ser integrado a ambiente virtuais/digitais de aprendizagem ou qualquer página da Internet, de código aberto e que disponha de recursos de acessibilidade, a pesquisa realizada proporcionou conhecimentos necessários para a escolha de um editor de texto, identificação de problemas de usabilidade e acessibilidade e o desenvolvimento de melhorias, que possibilitaram a utilização do editor por pessoas cegas, como comprovaram os testes realizados com usuários cegos. Para isto, baseado no método de investigação proposto e apresentando, identificou-se e desenvolveu-se novos recursos para o CKEditor, atendendo as recomendações da W3C para acessibilidade na Web.

Ressalta-se que profissionais que desenvolvem conteúdos para páginas de Internet devem seguir recomendações de acessibilidade para conteúdos Web avançados, na tentativa de garantir acessibilidade e uma melhor usabilidade para pessoas com deficiência, tornando os conteúdos Web disponíveis e a interação mais eficiente. Por outro lado, para que o usuário tenha maior produtividade ao utilizar aplicações Web, é necessário que também conheça tais recursos e recomendações de acessibilidades.

Algumas sugestões de melhorias foram surgindo, ao passo em que os objetivos deste trabalho eram atingidos. A experiência acerca dos recursos de acessibilidade que amadureceram durante a implementação e durante a realização dos testes com usuários para averiguação das melhorias, resultaram em algumas sugestões para trabalhos futuros, que são:

- Melhorar os recursos de acessibilidade utilizados na lista de tamanho da fonte do botão tamanho da fonte. Disponibilizar a frase “fonte 10” ao leitor de tela. Atualmente é disponibilizado apenas o número de tamanho da fonte, o que causou confusão para o usuário no teste. Nos testes com usuários a fim de averiguar os requisitos implementados, foi sugerido pelos usuários a melhoria da forma como o editor de texto interage com o leitor de tela NVDA, pois o mesmo não lê o texto quando o usuário leva o cursor para o início do texto ou quando a área de edição recebe o foco. Para esta melhoria, os editores de texto LibreOffice Writer e o Microsoft Word podem ser analisados e a mesma lógica de interação com o leitor de tela pode ser aplicada ao editor de texto para Web.
- Seguir padrões de teclas de atalho utilizadas por outros editores de texto. Torna-se necessário desenvolver um mecanismo que proporcione ao editor, quando recebe o foco, desabilitar as teclas de atalho configuradas para os navegadores e *plugins*, evitando conflitos. Neste trabalho não foi possível, por exemplo, utilizar a tecla *Ctrl+E* para centralizar o texto, pois esta tecla de atalho é utilizada pelo Firefox para utilizar a barra de pesquisa.
- Implementar o sistema de marcadores da mesma forma que o Microsoft Word implementa. Marcadores permitem efetuar hiperligações para outros pontos do documento, facilitando ao usuário navegar pelo documento. Esta funcionalidade foi

sugerida por um usuário durante a realização do teste de averiguação.

- Criar um comando acessível por tecla de atalho que realize uma audiodescrição da *interface* gráfica do editor de texto. Esta também é uma sugestão do usuário durante o teste de averiguação. Ele explicou que quando uma pessoa cega consegue montar uma imagem mental de como a *interface* está organizada, a navegação pela *interface* fica mais fácil. Também pode ser implementado um sistema de audiodescrição do local onde o mouse está posicionado, para tal, uma mensagem pode ser montada e enviada para o NVDA através da sua API (Application Programming Interface).
- Por fim, o propósito deste artigo estava voltado à experiência com o editor de texto Web, assim o método utilizado para a avaliação foi o de testes com os usuários. Como sugestão de trabalho futuro, indica-se a realização de testes sem o usuário, baseados em: heurísticas, guias de recomendação ou diretrizes como a ATAG 2.0 - Authoring Tool Accessibility Guidelines.

6. AGRADECIMENTOS

Núcleo de Informática na Educação Especial (NIEE) da Universidade Federal do Rio Grande do Sul (UFRGS) e Núcleo de Ação Comunitária da Universidade de Santa Cruz do Sul (NAC UNISC).

7. REFERÊNCIAS

- [1] BRASIL. (1996) *Decreto-Lei n. 9.394, de 20 de dezembro de 1996*. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/19394.htm>. Acesso em: mar. 2014.
- [2] CGI.BR. (2010) *Dimensões e características da Web brasileira: um estudo do .gov.br*. Disponível em <<http://www.cgi.br/media/docs/publicacoes/2/cgibr-nicbr-censoweb-govbr-2010.pdf>> Acesso em: mar. 2014.
- [3] eMAG. (2014) *Modelo de Acessibilidade em Governo Eletrônico*. Disponível em: <<http://emag.governoeletronico.gov.br/>>. Acesso em: mar. 2014.
- [4] Santarosa, L. M. C.; Conforto, D.; Basso, L. de O. (2010) AVA e ferramentas acessíveis: espaços de autoria coletiva e síncrona para a diversidade humana. In: *Congresso Iberoamericano de Informática Educativa - IE2010*, Santiago, Chile.
- [5] Santarosa, L. M. C. *et al.* (2007) Ambiente Virtual de Aprendizagem por Projetos – Edukito: Espaço para Inclusão Digital e Social de PNEES. In: *Revista Novas Tecnologias na Educação (RENTE)*, UFRGS - Porto Alegre, v.5, n. 1.
- [6] Nielsen, J.; Loranger, H. (2007) *Usabilidade na web: projetando websites com qualidade*. Rio de Janeiro: Campus.
- [7] Krug, S. (2006) *Não me faça pensar*. 2ed. Rio de Janeiro: Alta Books.
- [8] Dias, C. (2003) *Usabilidade na WEB Criando portais mais acessíveis*. Rio de Janeiro: Alta Books.

- [9] W3C. (2013) *Web Content Accessibility Guidelines* (WCAG). Disponível em: <<http://www.w3.org>>. Acessado em: mar. 2013.
- [10] CKSOURCE. (2013) *The best web text editor for everyone.*. Disponível em: <<http://www.ckeditor.com>>. Acesso em: mar. 2013.
- [11] Latour, B. (2007) Como falar do corpo? A dimensão normativa dos estudos sobre a ciência. In *J. A. Nunes & R. Roque (Orgs.) Objetos impuros. Experiências em estudos sociais da ciência*. Porto: Edições Afrontamento, p.39-61.
- [12] Despret, V. (2011) Os dispositivos experimentais. *Fractal, Revista Psicologia*, Rio de Janeiro, v.23, n.1. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1984-02922011000100004&lng=en&nrm=iso>. Acesso em: mar. 2014.
- [13] Viganico, T. E. (2013) *Editor de texto acessível para web*. Monografia (Graduação) - Universidade de Santa Cruz do Sul, 2013.
- [14] CGI.BR. (2011) *TIC Web – Relatório de Indicadores gov.br*. Disponível em: <<http://www.cgi.br/media/docs/publicacoes/4/cgibr-nicbr-w3c-censoweb-govbr-2011.pdf>>. Acesso em: 20 de ago. 2014.
- [15] IBGE. (2010) *Censo Demográfico 2010*. Disponível em: <<http://www.censo2010.ibge.gov.br>>. Acesso em: 30 mai. 2014.